1. A microprocessor, comprising:

an instruction buffer, comprising a plurality of stages for buffering instruction bytes received from an instruction cache;

a branch indicator associated with each of said plurality of stages, for storing an indication of whether or not the microprocessor branched to a target address of a branch instruction buffered in said associated stage; and

a multiplexer, coupled to said instruction buffer, for selecting one of said plurality of stages based on said branch indicator associated with one of said plurality of stages.

2. The microprocessor of claim 1, wherein said multiplexer provides a selected one of said plurality of stages to instruction format logic.

3. The microprocessor of claim 2, wherein said instruction buffer comprises a bottom stage of said plurality of stages.

4. The microprocessor of claim 3, wherein said instruction format logic determines a length of an instruction in said bottom stage.

5. The microprocessor of claim 4, wherein said multiplexer selects one of said plurality of stages based on said branch indicator associated with said bottom stage.

6. The microprocessor of claim 5, further comprising:

a wrap indicator, coupled to said multiplexer, for indicating whether or not said instruction wraps beyond said bottom stage, based on said instruction length.

7. The microprocessor of claim 6, wherein if said branch indicator indicates the microprocessor branched, then said multiplexer selects one of said plurality of stages based on whether said wrap indicator indicates said branch instruction wrapped beyond said bottom stage.

8. The microprocessor of claim 7, wherein if said branch indicator indicates the microprocessor branched, then if said wrap indicator indicates that said branch instruction wrapped beyond said bottom stage, then

said multiplexer selects one of said plurality of stages two above said bottom stage.

9. The microprocessor of claim 8, wherein said instruction buffer shifts out two of said plurality of stages.

10. The microprocessor of claim 8, wherein if said branch indicator indicates the microprocessor branched, then if said wrap indicator indicates that said branch instruction did not wrap beyond said bottom stage, then said multiplexer selects one of said plurality of stages one above said bottom stage.

11. The microprocessor of claim 10, wherein said instruction buffer shifts out one of said plurality of stages.

12. The microprocessor of claim 10, further comprising:

a carry indicator, coupled to said multiplexer, for indicating whether or not a byte of said instruction occupies a last byte of said bottom stage, based on said instruction length.

13. The microprocessor of claim 12, wherein if said branch indicator indicates the microprocessor did not branch,

then said multiplexer selects one of said plurality of stages based on whether or not said carry indicator indicates said instruction occupies a last byte of said bottom stage.

14. The microprocessor of claim 13, wherein if said branch indicator indicates the microprocessor did not branch, then if said carry indicator indicates said instruction occupies a last byte of said bottom stage, then said multiplexer selects one of said plurality of stages one above said bottom stage.

15. The microprocessor of claim 14, wherein said instruction buffer shifts out one of said plurality of stages.

16. The microprocessor of claim 14, wherein if said branch indicator indicates the microprocessor did not branch, then if said carry indicator indicates said instruction does not occupy a last byte of said bottom stage, then said multiplexer selects said bottom stage.

17. The microprocessor of claim 16, wherein said instruction buffer shifts out none of said plurality of stages.

18. The microprocessor of claim 1, further comprising:

a branch target address cache, coupled to receive a fetch address of said instruction cache, for providing said target address of said branch instruction as a subsequent fetch address to said instruction cache.

19. The microprocessor of claim 18, wherein said branch target address cache provides an indication of whether or not said instruction cache fetch address hit in said branch target address cache.

20. The microprocessor of claim 19, further comprising:

control logic, coupled to said multiplexer, for populating said branch indicator based on said indication of whether or not said fetch address hit in said branch target address cache.

21. A pre-decode stage in a microprocessor, comprising:

an instruction buffer, comprising at least stages A, B, and C, for buffering instruction bytes;

a multiplexer, coupled to said instruction buffer, for selecting one of said stages A, B, and C for provision to instruction format logic;

a branch indicator, coupled to said multiplexer, for indicating whether the microprocessor branched based on a branch instruction present in said stage A; and

a wrap indicator, coupled to said multiplexer, for indicating whether an instruction formatted by said instruction format logic wraps across said stages A and B;

wherein said multiplexer selects one of said stages A, B, and C for provision to said instruction format logic based on said branch indicator and said wrap indicator.

22. The pre-decode stage of claim 21, further comprising an adder, coupled to said multiplexer, said adder comprising:

a first input, for receiving a pointer to a location of said instruction within said stage A;

a second input, for receiving a length of said instruction from said instruction format logic;

a first output, for specifying a sum of said first and second inputs; and

a second output, for indicating a carry of said sum.

23. The pre-decode stage of claim 22, wherein said wrap indicator indicates true only if said carry is true and said sum is not zero.

24. The pre-decode stage of claim 23, wherein said multiplexer selects stage C if said branch indicator indicates true and said wrap indicator indicates true.

25. The pre-decode stage of claim 24, wherein said instruction buffer shifts out said stages A and B after said multiplexer selects stage C.

26. The pre-decode stage of claim 23, wherein said multiplexer selects stage B if said branch indicator indicates true and said wrap indicator indicates false.

27. The pre-decode stage of claim 26, wherein said instruction buffer shifts out said stage A after said multiplexer selects stage B.

28. The pre-decode stage of claim 23, wherein said multiplexer selects stage B if said branch indicator indicates false and said second output indicates true.

29. The pre-decode stage of claim 28, wherein said instruction buffer shifts out said stage A after said multiplexer selects stage B.

30. The pre-decode stage of claim 23, wherein said multiplexer selects stage A if said branch indicator indicates false and said second output indicates false.

31. The pre-decode stage of claim 30, wherein said instruction buffer shifts out none of said stages A, B, and C after said multiplexer selects stage A.

32. The pre-decode stage of claim 21, wherein said multiplexer also provides to said instruction format logic along with said selected one of said stages A, B, and C a shadowed portion of one of said stages A, B, and C immediately above said selected one of said stages A, B, and C.

33. The pre-decode stage of claim 32, wherein said instruction comprises a variable length instruction.

34. The pre-decode stage of claim 33, wherein said variable length instruction comprises an x86 instruction.

35. The pre-decode stage of claim 34, wherein said shadowed portion comprises at least 10 instruction bytes.

36. A microprocessor branch control apparatus, comprising:

an instruction buffer, comprising first, second, and third stages for buffering first, second, and third cache lines received from an instruction cache, said first and second cache lines each containing a portion of a branch instruction, said third cache line containing a target instruction of said branch instruction;

a branch target address cache (BTAC), coupled to said instruction buffer, for outputting an indication that said third cache line was selected from said instruction cache by a target address of said branch instruction provided by said BTAC; and

a multiplexer, coupled between said instruction buffer and instruction format logic, for selecting one of said first, second, and third stages for provision to said instruction format logic;

wherein said multiplexer selects said third stage, after selecting said first stage, based on said indication output by said BTAC and based on a length of said branch instruction determined by said instruction format logic.

37. The branch control apparatus of claim 36, further comprising:

a pointer into said first stage, for specifying a starting location of said branch instruction.

38. The branch control apparatus of claim 37, further comprising:

an adder, for generating a sum of said pointer and said length of said branch instruction;

wherein said sum indicates that said first and second cache lines each contain a portion of said branch instruction.

39. The branch control apparatus of claim 38, wherein said multiplexer selects said third cache line based on said sum of said pointer and said length of said branch instruction and said indication output by said BTAC.

40. The branch control apparatus of claim 38, further comprising:

a second multiplexer, coupled to said pointer, for selecting one of said sum and a portion of said target address as a next one of said pointer.

41. A method for buffering instruction bytes for provision to instruction format logic in a microprocessor, the method comprising:

storing an indication of whether or not the processor branched in response to a first cache line stored in an instruction buffer;

generating a length of a first instruction in said first cache line stored in said instruction buffer;

determining whether said first instruction wraps beyond said first cache line based on said length of said first instruction; and

selecting a second cache line stored in said instruction buffer for formatting a second instruction, based on said indication and said determining.

42. The method of claim 41, wherein said second cache line is two cache lines above said first cache line if said indication indicates the processor branched and said first instruction wraps beyond said first cache line.

43. The method of claim 42, further comprising:

shifting said first cache line and a cache line above said first cache line out of said instruction buffer, after said selecting.

44. The method of claim 41, wherein said second cache line is one cache line above said first cache line if said indication indicates the processor branched and said first instruction does not wrap beyond said first cache line.

45. The method of claim 44, further comprising:

shifting said first cache line out of said instruction buffer, after said selecting.

46. The method of claim 41, further comprising:

determining whether a byte in said first instruction occupies a last byte of said first cache line in said instruction buffer.

47. The method of claim 46, wherein said second cache line is one cache line above said first cache line if said indication indicates the processor did not branch and a byte in said first instruction occupies a last byte of said first cache line.

48. The method of claim 47, further comprising:

shifting said first cache line out of said instruction buffer, after said selecting.

49. The method of claim 46, wherein said second cache line is said first cache line if said indication indicates the processor did not branch and a byte in said first instruction does not occupy a last byte of said first cache line.

50. The method of claim 49, further comprising:

shifting no cache lines out of said instruction buffer, after said selecting.